

Improved Diffuse Boundary Condition for the DSBGK Method to Eliminate the Unphysical Density Drift

Jun Li^a

^a*Applied Mathematics and Computational Science
King Abdullah University of Science and Technology
Thuwal, Saudi Arabia*

Abstract

An improved diffuse boundary condition, where the number flux of the incoming real molecules on the wall surface is calculated using the molecular variables rather than the cell's macroscopic variables, is proposed to eliminate the unphysical density drift, which was observed in the previous DSBGK simulation of the lid-driven problem but disappears in the channel flow problem because of the density constraint imposed at the open boundaries. Consequently, the efficient time-average process is valid for sampling all quantities of interest in closed as well as open problems. When the driven velocity of the lid-driven problem is only one micrometer per second that is realistic in the micro-electro-mechanical systems or pore-scale flows of the shale gas, we suggest to use the original boundary condition because the density drift during a very long time interval becomes unperceivable when the perturbation is tiny. Thus, the original diffuse boundary condition, which contains much less stochastic noise than the improved one, is still an advisable choice in closed problems with tiny perturbation and in open problems unless the density drift occurs and fails the goal of the simulations making the use of the improved boundary condition necessary.

Keywords: rarefied gas flows, micro gas flows, pore-scale flows of shale gas, Boltzmann equation, BGK equation, molecular simulation methods, DSMC method, variance reduction, surface reflection.

e-mail: lijun04@gmail.com

To view the network videos of the DSBGK transient results of several benchmark problems, please click the *Dropbox link* or find the link in *CV*.

1. Introduction

For micro gas flows, the Boltzmann equation rather than the Navier-Stokes equation should be used due to the high Knudsen number $Kn = \lambda/L$ where λ is the molecular mean free path and L is the characteristic length of the flow problem. The characteristic velocity of micro gas flows is usually much smaller than the molecular random thermal velocity and sometimes the variations of quantities of interest inside the flow domain are very small, which makes the computational cost of the traditional DSMC method [1] prohibitive although it is efficient in the case of high-speed flows. The design of the micro-electro-mechanical systems (MEMS) promoted the development of numerical simulation methods devoted to the low-speed problems of high Kn . Recently, the simulations of pore-scale flows of the shale gas, where the pore size is about 10-100 nm and the flow velocity is tiny, are desirable since the experimental measures to get the permeability of representative rock samples may take a couple of months.

The DSBGK method, which is proposed in [2] and detailed in [3], is very efficient for simulating low-speed problems of high Kn and verified against the DSMC method in the lid-driven, Couette, channel flow and thermal transpiration problems over a wide range of Kn [3]. In addition to its high-efficiency, the DSBGK method has many numerical advantages including simplicity, stability, convenience for complex configuration and for parallel computation because the basic algorithmic structure of the DSMC method is employed.

In the original diffuse boundary condition, we use the cell's variables to compute the number flux of incoming real molecules on the wall surface such that the transient results contain low stochastic noise. This also reduces the memory usage in open problems where the DSBGK simulation remains stable when using only 10 simulated molecules per cell [3]. But, in closed problems, the density drift occurs due to the lack of density constraint that is imposed at the open boundaries of open problems. Although the DSBGK simulations of closed problems using about 10 simulated molecules per cell are also stable (see Fig. 4), the density drift prohibits the application of the time-average process used to reduce the stochastic noise which is obvious when using such small amount of simulated molecules. We propose an improved boundary condition here to eliminate the unphysical density drift and then the efficient time-average process is valid for sampling all quantities of interest in closed as well as open problems. The DSBKG steady state results obtained using

the time-average process are verified in the lid-driven problem as an example.

2. DSBGK Method

We consider the gas flows of single component. In the absence of external body force, the BGK equation [4] can be written as a Lagrangian form:

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + c_j \frac{\partial f}{\partial x_j} = v(f_{\text{eq}} - f), \quad (1)$$

where $f(t, \vec{x}, \vec{c})$ is the unknown probability distribution function, t the time, \vec{x} the spatial coordinate, \vec{c} the molecular velocity and, the parameter v is selected appropriately to satisfy the coefficient of dynamic viscosity μ or heat conduction κ as follows [7]:

$$\begin{cases} \mu_{\text{BGK}} = \frac{nk_{\text{B}}T}{v} \\ \kappa_{\text{BGK}} = \frac{5k_{\text{B}}}{2m} \frac{nk_{\text{B}}T}{v}, \end{cases} \quad (2)$$

and the Maxwell distribution function f_{eq} is:

$$f_{\text{eq}}(t, \vec{x}, \vec{c}) = n \left(\frac{m}{2\pi k_{\text{B}}T} \right)^{3/2} \exp \left[\frac{-m(\vec{c} - \vec{u})^2}{2k_{\text{B}}T} \right], \quad (3)$$

where m is the molecular mass, k_{B} the Boltzmann constant and, the number density n , flow velocity $\vec{u} = (u_x, u_y, u_z)$ and temperature T are functions of t and \vec{x} and defined using the integrals of f with respect to \vec{c} .

The DSBGK method is proposed in [2] and detailed in [3] where the extension to problems with external force is discussed. The simulation process is divided into a series of time steps Δt and the computational domain is divided into many cells. Δt and the cell size Δx are selected the same as in the DSMC method when simulating problems of high Kn . Each simulated molecule l carries four molecular variables: position \vec{x}_l , molecular velocity \vec{c}_l , number N_l of real molecules represented by the simulated molecule l , and F_l which is equal to $f(t, \vec{x}_l, \vec{c}_l)$. The variables $n_{\text{tr},k}$, $\vec{u}_{\text{tr},k}$, $T_{\text{tr},k}$ of each cell k are updated using \vec{x}_l , \vec{c}_l and the increment of N_l based on the mass, momentum and energy conservation principles of the intermolecular collision process and are simultaneously used to update the molecular variables based on the BGK equation and the extrapolation [3] of acceptance-rejection scheme.

The DSBGK method is a molecular simulation method and theoretically all macroscopic quantities of interest should be computed using the molecular variables. The cell's variables $n_{\text{tr},k}, \vec{u}_{\text{tr},k}, T_{\text{tr},k}$ are auxiliary variables, which replace the original macroscopic quantities n_k, \vec{u}_k, T_k defined by the molecular variables to reduce the stochastic noise, and converge to n_k, \vec{u}_k, T_k as discussed after Eq. (13) of [3].

2.1. Summary of the DSBGK algorithm

1. Initialization. Generate many cells and simulated molecules and assign them initial values for $n_{\text{tr},k}, \vec{u}_{\text{tr},k}, T_{\text{tr},k}$ and $\vec{x}_l, \vec{c}_l, F_l, N_l$, respectively.

2. Each simulated molecule moves uniformly and in a straight line before encountering boundaries. During each Δt , the trajectory of each particular molecule l may be divided into several segments by the cell's interfaces. Then, \vec{x}_l, F_l, N_l are *deterministically* updated along each segment in sequence. When encountering the wall boundaries, \vec{c}_l is randomly updated according to the reflection model and then F_l is updated correspondingly. In open problems, simulated molecules are removed from the computational domain when moving across the open boundaries during each Δt and new simulated molecules are generated at the open boundaries after each Δt . The variables $n_{\text{tr},k}, \vec{u}_{\text{tr},k}, T_{\text{tr},k}$ of each cell k are updated after each Δt .

3. After convergence, $n_{\text{tr},k}, \vec{u}_{\text{tr},k}, T_{\text{tr},k}$ are used as the discrete solutions of the BGK equation at steady state.

2.2. Wall boundary conditions

In the reflection models of wall boundary, \vec{c}_l and then F_l are changed after molecular reflection at \vec{x}_l on the wall. N_l remains unchanged to conserve the mass. The reflecting velocity \vec{c}_r is randomly selected the same as in the DSMC method and then \vec{c}_l is updated to $\vec{c}_r + \vec{u}_{\text{wall}}$ where \vec{u}_{wall} is the wall velocity (see the details in [3]). The subscript l is omitted in the notations of the incoming and reflecting velocities.

As discussed in [3], F_l is updated to $F_{l,\text{new}} = f(t, \vec{x}_l, \vec{c}_{l,\text{new}}) = f(t, \vec{x}_l, \vec{c}_r + \vec{u}_{\text{wall}})$ after getting \vec{c}_r . Note that F_l is the representative value of f which is different from the scatter kernel R used to select \vec{c}_r for each particular reflection process. Generally speaking, f is related to the mass flux but R has nothing to do with the mass flux. Usually, R describes the probability distribution of \vec{c}_r inside the half velocity space ($\vec{c}_r \cdot \vec{e}_n > 0$ where \vec{e}_n is the outer-normal unit vector of the wall) as a function depending on the wall temperature T_{wall} , accommodation coefficients α_n, α_τ and possibly also on the

incoming velocity \vec{c}_i . So, we have $R = R(\vec{c}_r, \vec{c}_i)$ which contains $T_{\text{wall}}, \alpha_n, \alpha_\tau$ as parameters. R satisfies the normalization condition $\int_{\vec{c}_r \cdot \vec{e}_n > 0} R(\vec{c}_r, \vec{c}_i) d\vec{c}_r = 1$ where $R(\vec{c}_r, \vec{c}_i) d\vec{c}_r$ is the probability for the molecule coming at \vec{c}_i to reflect into the velocity space element $d\vec{c}_r$ around \vec{c}_r . The transformation between f at the wall boundary and R can be completed using the incoming mass flux.

We introduce $f_B(\vec{c})$ as the equivalent distribution function of f observed at the reflection point \vec{x}_l and at the current moment t in a local Cartesian reference frame S_{local} moving at \vec{u}_{wall} , which means $f_B(\vec{c}) = f(t, \vec{x}_l, \vec{c} + \vec{u}_{\text{wall}})$. After getting the formula of $f_B(\vec{c})$, we have $F_{l,\text{new}} = f_B(\vec{c}_r)$. The distribution $f_B(\vec{c}_i)|_{\vec{c}_i \cdot \vec{e}_n < 0}$ of the incoming molecules is known from the molecular information in the adjacent cell. $f_B(\vec{c}_r)|_{\vec{c}_r \cdot \vec{e}_n > 0}$ is the distribution of reflecting molecules and related to R as introduced in [8]:

$$f_B(\vec{c}_r)(\vec{c}_r \cdot \vec{e}_n) d\vec{c}_r = - \int_{\vec{c}_i \cdot \vec{e}_n < 0} R(\vec{c}_r, \vec{c}_i) f_B(\vec{c}_i)(\vec{c}_i \cdot \vec{e}_n) d\vec{c}_i d\vec{c}_r. \quad (4)$$

Taking integration of Eq. (4) with respect to \vec{c}_r over its half velocity space and using the normalization condition of $R(\vec{c}_r, \vec{c}_i)$, we get:

$$\begin{aligned} & \int_{\vec{c}_r \cdot \vec{e}_n > 0} f_B(\vec{c}_r)(\vec{c}_r \cdot \vec{e}_n) d\vec{c}_r \\ &= - \int_{\vec{c}_r \cdot \vec{e}_n > 0} \int_{\vec{c}_i \cdot \vec{e}_n < 0} R(\vec{c}_r, \vec{c}_i) f_B(\vec{c}_i)(\vec{c}_i \cdot \vec{e}_n) d\vec{c}_i d\vec{c}_r \\ &= - \int_{\vec{c}_i \cdot \vec{e}_n < 0} f_B(\vec{c}_i)(\vec{c}_i \cdot \vec{e}_n) d\vec{c}_i, \end{aligned} \quad (5)$$

which represents the mass conservation of molecular reflection process.

In the Maxwell diffuse reflection model, we have:

$$f_{B,\text{diffuse}}(\vec{c}_r) = n_{\text{eff}} \left(\frac{m}{2\pi k_B T_{\text{wall}}} \right)^{3/2} \exp\left(\frac{-m\vec{c}_r^2}{2k_B T_{\text{wall}}} \right), \quad (6)$$

where the effective n_{eff} is determined from $f_B(\vec{c}_i)$ based on the mass conservation principle of Eq. (5). Theoretically, $f_B(\vec{c}_i)$ depends on the incoming molecules. To reduce the stochastic noise, we previously use the cell's variables rather than the molecular variables to determine $f_B(\vec{c}_i)$ as follows:

$$f_{B,\text{bad}}(\vec{c}_i) = n_{\text{tr},k} \left(\frac{m}{2\pi k_B T_{\text{tr},k}} \right)^{3/2} \exp\left[\frac{-m(\vec{c}_i - (\vec{u}_{\text{tr},k} - \vec{u}_{\text{wall}}))^2}{2k_B T_{\text{tr},k}} \right], \quad (7)$$

where $n_{\text{tr},k}$, $\vec{u}_{\text{tr},k}$, $T_{\text{tr},k}$ are the quantities of cell k close to the reflection point \vec{x}_l . Then, the number of incoming real molecules on per unit wall area during per unit time is:

$$\begin{aligned} N_{\text{in,bad}} &= - \int_{\vec{c}_i \cdot \vec{e}_n < 0} f_{\text{B,bad}}(\vec{c}_i)(\vec{c}_i \cdot \vec{e}_n) d\vec{c}_i \\ &= n_{\text{tr},k} \sqrt{\frac{k_B T_{\text{tr},k}}{2\pi m}} [\exp(-u_{\text{in}}'^2) + \sqrt{\pi} u_{\text{in}}' (1 + \text{erf}(u_{\text{in}}'))], \end{aligned} \quad (8)$$

where $u_{\text{in}}' = \frac{-(\vec{u}_{\text{tr},k} - \vec{u}_{\text{wall}}) \cdot \vec{e}_n}{\sqrt{2k_B T_{\text{tr},k}/m}}$. Similarly, the number N_{out} of reflecting real molecules is:

$$N_{\text{out}} = \int_{\vec{c}_r \cdot \vec{e}_n > 0} f_{\text{B,diffuse}}(\vec{c}_r)(\vec{c}_r \cdot \vec{e}_n) d\vec{c}_r = n_{\text{eff}} \sqrt{\frac{k_B T_{\text{wall}}}{2\pi m}}. \quad (9)$$

Let $N_{\text{out}} = N_{\text{in,bad}}$ as required by Eq. (5), we get an estimation of n_{eff} :

$$n_{\text{eff,bad}} = n_{\text{tr},k} \sqrt{\frac{T_{\text{tr},k}}{T_{\text{wall}}}} [\exp(-u_{\text{in}}'^2) + \sqrt{\pi} u_{\text{in}}' (1 + \text{erf}(u_{\text{in}}'))]. \quad (10)$$

After getting $n_{\text{eff,bad}}$, we can determine $F_{l,\text{new}} = f_{\text{B,diffuse}}(\vec{c}_r)$ where $n_{\text{eff}} = n_{\text{eff,bad}}$. We store $n_{\text{eff,bad}}$ and repeatedly use it for different simulated molecules reflecting on the same subarea adjacent to the cell k during the same Δt and update $n_{\text{eff,bad}}$ after each Δt .

The applications of the CLL reflection model [5]-[6] and the specular reflection model are discussed in [3].

3. Improved Diffuse Reflection Model

To update F_l after reflecting on the wall, the diffuse and CLL reflection models need the distribution $f_{\text{B}}(\vec{c}_i)$ of all incoming molecules but the specular reflection model only uses the information of each particular incoming molecule. In the above diffuse reflection model, we use Eq. (7) as $f_{\text{B}}(\vec{c}_i)$ for simplicity to calculate the incoming number flux such that the stochastic noise is reduced. This simplification is reasonable when $f_{\text{B}}(\vec{c})$ near the boundary is close to the local Maxwell distribution where the incoming number flux depends only on n , \vec{u} and T . But, the deviation of $f_{\text{B}}(\vec{c})$ from the Maxwell distribution usually originates from the boundary and could be large at the boundary. The accuracy of Eq. (7) depends on the magnitude of deviation.

Actually, the incoming number flux can be directly calculated using the information of incoming molecules although the correspondingly computed flux contains high stochastic noise since the number of simulated molecules per cell is usually small. As in the DSMC method, it is convenient for the DSBGK method to calculate the *net* flux $\Gamma(Q)$ of any molecular quantity $Q(\vec{c})$ in unit time and across unit area of the boundary surface:

$$\Gamma(Q) = \frac{1}{\Delta t \Delta S} \sum_l N_l [Q(\vec{c}_i) - Q(\vec{c}_r)]_l, \quad (11)$$

where the summation is over the simulated molecules reflecting on the sub-area ΔS during the current time step Δt , $Q(\vec{c}_i)$ and $Q(\vec{c}_r)$ are the incoming and reflecting quantities, respectively. Let $Q = m\vec{c}$ and $m\vec{c}^2/2$ and then $\Gamma(Q)$ represents the stress and heat flux, respectively. Similarly, the incoming number flux is computed as:

$$N_{\text{in}} = \frac{1}{\Delta t \Delta S} \sum_l N_l. \quad (12)$$

Let $N_{\text{out}} = N_{\text{in}}$ as required by Eq. (5), we get an improved formula for n_{eff} :

$$n_{\text{eff}} = \sqrt{\frac{2\pi m}{k_B T_{\text{wall}}}} \frac{1}{\Delta t \Delta S} \sum_l N_l. \quad (13)$$

$F_{l,\text{new}} = f_{\text{B,diffuse}}(\vec{c}_r)$ is implemented to update F_l using Eq. (6) during the simulation process and n_{eff} is updated using Eq. (13) after each Δt . The division of ΔS is based on the cell's division and each of those cell's interfaces located at the wall surfaces has a variable n_{eff} . At the initial state with Maxwell distribution and $(\vec{u}_{\text{tr},k} - \vec{u}_{\text{wall}}) \cdot \vec{e}_n = 0$, we have $n_{\text{eff}}|_{t=0} = n_{\text{tr},k} \sqrt{\frac{T_{\text{tr},k}}{T_{\text{wall}}}}$. We note that $N_{\text{in,bad}}$ is of the same order of magnitude as the expected value of N_{in} since the incoming number flux depends mostly on the macroscopic quantities, which accounts for the favourable simulation results obtained using $N_{\text{in,bad}}$ [3]. Although the use of the improved boundary condition leads to negligible increase of the computational time for each time step, more samples are required to reduce the stochastic noise of average results because N_{in} contains more stochastic noise than $N_{\text{in,bad}}$ as implied in Fig. 3.

In addition to the small improvement of accuracy (see the above discussion and compare Fig. 2 with Fig. 5 of [3]), another advantage (see Fig. 3)

of using the molecular variables to compute the incoming number flux by Eq. (12) is to eliminate the unphysical drift of $n_{\text{tr},k}$, which was observed in the previous simulations of *closed* problems using $N_{\text{in,bad}}$ and disappears in the open problems because of the fixed density imposed at the open boundaries [3]. We didn't observe drift in the distributions of $\vec{u}_{\text{tr},k}$ and $T_{\text{tr},k}$, which implies that the velocity \vec{u}_k and temperature T_k defined using the molecular variables as in Eq. (3) of [3] are averagely equal to $\vec{u}_{\text{tr},k}$ and $T_{\text{tr},k}$, respectively (see the discussion after Eq. (13) of [3]). In order to simplify the analysis, we assume that $\vec{u}_{\text{tr},k} = \vec{u}_{\text{wall}}$, $T_{\text{tr},k} = T_{\text{wall}}$ and the distribution of simulated molecules is the local Maxwell distribution in each cell k adjacent to the wall. Thus, N_{in} computed by Eq. (12) is averagely equal to $n_k \sqrt{\frac{k_B T_k}{2\pi m}}$ and n_{eff} computed by Eq. (13) is averagely equal to $n_k = \sum N_l / V_k$ where $\sum N_l$ is the total number of real molecules in the cell k and V_k is the volume of cell k . When $n_k < n_{\text{tr},k}$, n_{eff} is smaller than $n_{\text{tr},k}$ and thus, during the intermolecular collision process in the cell k , $F_{l,\text{new}} = f_{\text{B,diffuse}}(\vec{c}_r)$ after reflecting on the wall computed by Eq. (6) is smaller than $f_{\text{eq,tr},k}(\vec{c}_{l,\text{new}})$ making the corresponding $\Delta_k N_l$ positive, which provides a mechanism to increase n_k and decrease $n_{\text{tr},k}$ as required by the convergence where $n_k = n_{\text{tr},k}$ is satisfied on average. In contrast, if the incoming number flux is computed by Eq. (8) using the cell's variables, we have $n_{\text{eff}} = n_{\text{eff,bad}} = n_{\text{tr},k}$ according to Eq. (10) and then, during the intermolecular collision process in the cell k , $F_{l,\text{new}}$ after reflecting on the wall is equal to $f_{\text{eq,tr},k}(\vec{c}_{l,\text{new}})$ making the previous mechanism of convergence disappeared. Similar analysis applies to the case of $n_k > n_{\text{tr},k}$. In real applications where the above assumptions for simplicity of analysis are not true, the use of actual incoming number flux computed by Eq. (12) also improves the numerical performance with respect to convergence because: 1) n_{eff} computed by Eq. (13) using the molecular variables is close to n_k on average even though the molecular distribution near the wall deviates from the Maxwell distribution and $\vec{u}_k = \vec{u}_{\text{tr},k} \neq \vec{u}_{\text{wall}}$, $T_k = T_{\text{tr},k} \neq T_{\text{wall}}$; 2) the magnitudes of $F_{l,\text{new}} = f_{\text{B,diffuse}}(\vec{c}_r)$ and $f_{\text{eq,tr},k}(\vec{c}_{l,\text{new}})$ depend mostly on n_{eff} and $n_{\text{tr},k}$, respectively, since $\vec{u}_{\text{tr},k} - \vec{u}_{\text{wall}}$ and $T_{\text{tr},k} - T_{\text{wall}}$ are relatively small.

4. Validate the Improved Diffuse Reflection Model

In DSBGK simulations, the parameter v is selected to satisfy the coefficient of dynamic viscosity μ or heat conduction κ by Eq. (2). For problems where the momentum exchange is the dominant effect, we use

$v = v(\mu) \equiv nk_{\text{B}}T/\mu$ to satisfy μ . For problems where the heat conduction is the dominant effect, we select $v = v(\kappa) \equiv 5nk_{\text{B}}^2T/(2m\kappa)$ to satisfy κ . Note that μ is given usually. For monoatomic gas where the Prandtl number is $Pr = 2/3$ and the specific heat capacity at constant pressure is $C_p = 5k_{\text{B}}/(2m)$, we have $v(\kappa) = 2nk_{\text{B}}T/(3\mu)$ as $\kappa = C_p\mu/Pr$. Additionally, we define the Knudsen number $Kn = \lambda/L$ of the problem concerned using the initial molecular mean free path $\lambda_0 = \frac{16\mu}{5n_0\sqrt{2\pi mk_{\text{B}}T_0}}$ [8], where n_0 and T_0 are the initial density and temperature, respectively.

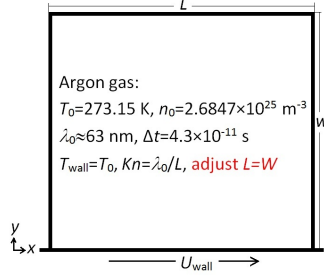


Figure 1: Schematic model of the lid-driven cavity flow.

We choose the lid-driven problem of $Kn = 6.3$ and $U_{\text{wall}} = 20$ m/s, where the density drift was observed in the previous DSBGK simulation which converges after about $40\Delta t$ taking about 11 minutes of CPU time running on one processor of Lenovo laptop E43A and has obvious deviation of density after $900\Delta t$ (see Fig. 5 of [3]), to verify the validity of the improved boundary condition and to show the elimination of the density drift. As in the previous DSBGK simulation, we use $v = v(\mu)$, the cell number is 20×20 and each cell contains about 2000 simulated molecules on average. The DSBGK simulation process converges also after about $40\Delta t$, which takes about 5 minutes of CPU time running on one processor of MacBook Air. The difference of CPU time between the previous and current simulations is mostly due to using different laptops. We collect 30 samples sampled once at the end of each Δt after the initial $40\Delta t$. The total computational time is about 9 minutes of CPU time. Fig. 2 gives the validation of the DSBGK time-average results against the DSMC time-average results. The agreement between the DSBGK and DSMC methods in Fig. 2 is excellent and slightly better than that in Fig. 5 of [3] where the agreement is very good though.

We compute the transient total number of real molecules carried by all

simulated molecules, namely $\sum_{\text{Domain}} N_l$, and the total number represented by all cells, namely $\sum_{\text{Domain}} n_{\text{tr},k} V_k$. The evolutions of this two quantities normalized by the initial total number $\sum_{\text{Domain}}^{\text{init}} N_l$ of real molecules is given in Fig. 3 which clearly shows that the unphysical density drift disappears after using the improved boundary condition. Thus, the efficient time-average process is valid for sampling all quantities in close as well as open problems when using the improved boundary condition. Fig. 3 also shows that the stochastic noise of the transient results is significantly increased when replacing the original boundary condition by the improved one which is more accurate but has more stochastic noise than the original one.

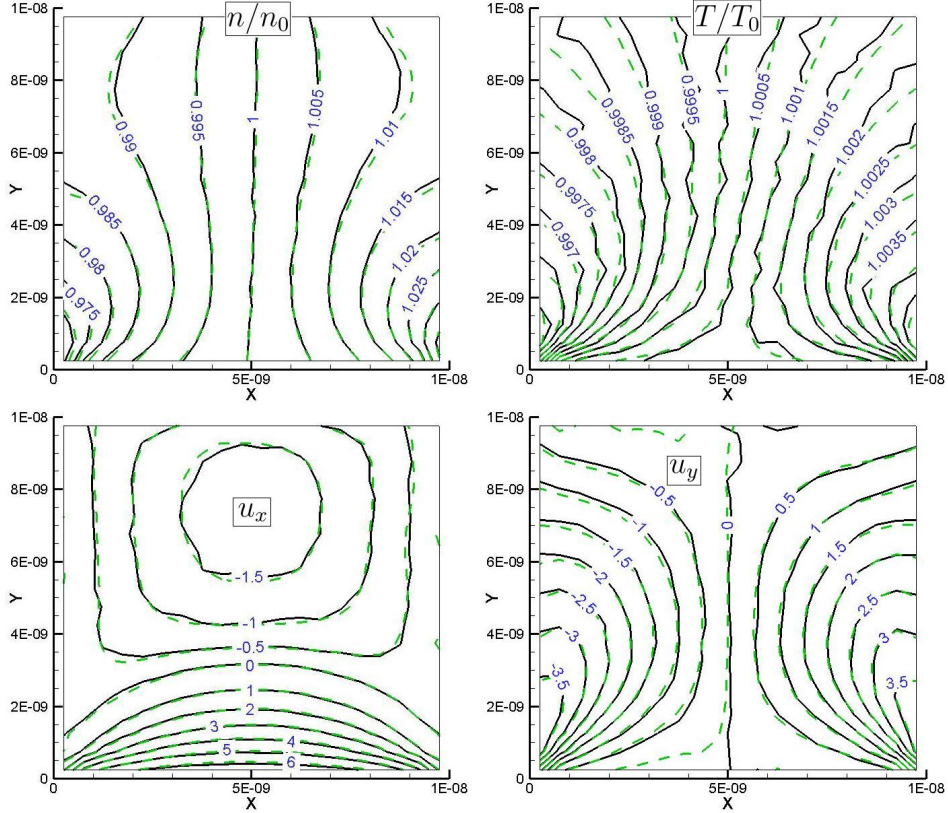


Figure 2: The steady state results of the lid-driven problem, $Kn = 6.3$ and $U_{\text{wall}} = 20$ m/s; Dashed lines: DSBGK time-average results using 2000 simulated molecules per cell and 30 samples and totally taking about 9 minutes of CPU time, Solid lines: DSMC time-average results taking more than one day of CPU time.

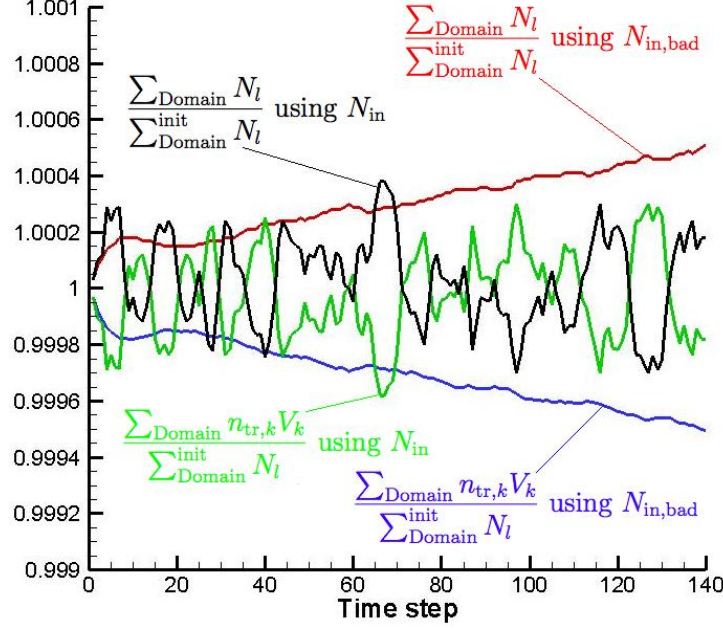


Figure 3: Evolutions of the total numbers of real molecules represented by all simulated molecules and all cells, respectively, in the DSBGK simulations of the lid-driven problem using two different boundary conditions, $Kn = 6.3$ and $U_{\text{wall}} = 20$ m/s, 2000 simulated molecules per cell.

5. Use the Original Diffuse Reflection Model in Problems with Tiny Perturbation

To show the high efficiency of the DSBGK method in simulating low-speed problems, we model the lid-driven problem again and only change the driven velocity U_{wall} from 20 m/s to 10^{-6} m/s which is realistic in the MEMS or pore-scale flows of the shale gas where the pore size is about 10-100 nm. Unfortunately, the transient DSBGK results obtained using the improved boundary condition based on N_{in} are dominated by the stochastic noise. Thus, we suggest to use the original boundary condition based on $N_{\text{in,bad}}$, which has much less noise than N_{in} , because the difference between $N_{\text{in,bad}}$ and the expected value of N_{in} is proportional (although may not linear) to the magnitude of perturbation which is tiny here. $n_{\text{eff,bad}}$ computed using $N_{\text{in,bad}}$ is almost exactly equal to the initial uniform number density n_0 , which is physically accurate when the perturbation of the closed problem is tiny. In the DSBGK simulation using the original boundary condition, we use 2000

simulated molecules per cell again and thus the computational time for the initial $40\Delta t$ is almost the same as in the previous case with $U_{\text{wall}} = 20$ m/s, namely about 5 minutes of CPU time. The simulation process converges also after about $40\Delta t$ and the density drift at the $900^{\text{th}}\Delta t$ is unperceivable as shown in Fig. 4, where the DSBGK results at the $40^{\text{th}}\Delta t$ obtained using only 10 simulated molecules per cell are given together to show the numerical stability of using few simulated molecules in closed problems.

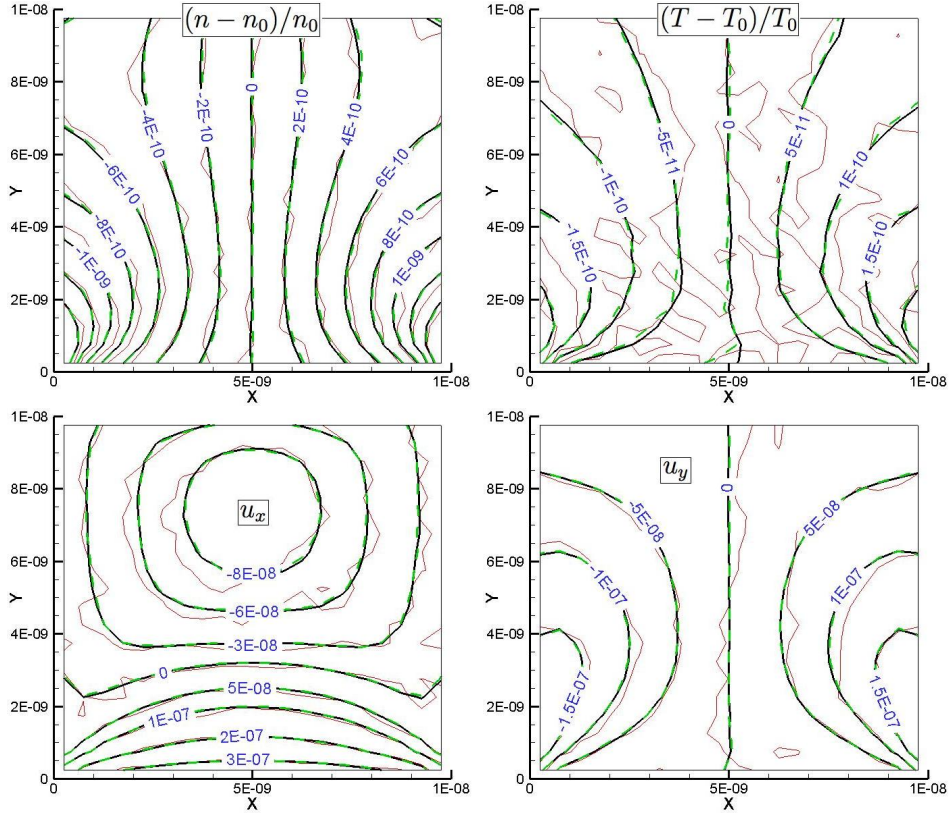


Figure 4: The steady state results (without average) of the DSBGK simulations of the lid-driven problem using the *original* boundary condition, $Kn = 6.3$ and $U_{\text{wall}} = 10^{-6}$ m/s; Dashed green lines: results using 2000 simulated molecules per cell at the $40^{\text{th}}\Delta t$ taking about 5 minutes of CPU time, Solid black lines: results using 2000 simulated molecules per cell at the $900^{\text{th}}\Delta t$, Thin solid red lines: results using 10 simulated molecules per cell at the $40^{\text{th}}\Delta t$ (DSMC simulation theoretically requires $(\frac{20}{10^{-6}})^2$ days).

6. Conclusions

An improved diffuse boundary condition based on N_{in} is proposed to eliminate the density drift observed in the previous DSBGK simulation of *closed* problem using the original boundary condition based on $N_{\text{in,bad}}$. Consequently, the efficient time-average process is valid in the DSBGK simulations of closed as well as open problems.

The unphysical density drift becomes unperceivable when the perturbation is tiny. Thus, for closed problems with tiny perturbation, we suggest to use the original boundary condition which makes the transient results smooth. In open problems, the original boundary condition is always an advisable choice to reduce the memory usage by using few simulated molecules. We don't need to resort to the improved boundary condition unless the density drift occurs and fails the goal of the simulations.

References

- [1] Graeme A. Bird, Molecular Gas Dynamics and the Direct Simulation of Gas Flows, Clarendon Press, Oxford, 1994.
- [2] Jun Li, Direct Simulation Method Based on BGK Equation, in: 27th International Symposium on Rarefied Gas Dynamics, AIP, 2011, 283–288 (*presented first in ESPCI, Paris, 2009*).
- [3] Jun Li, Comparison between the DSMC and DSBGK Methods, arXiv:1207.1040 [physics.comp-ph], 2012.
- [4] P. L. Bhatnagar, E. P. Gross, M. Krook, A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems, Physical Review 94(3) (1954) 511–525.
- [5] Carlo Cercignani, Maria Lampis, Kinetic Models for Gas-surface Interactions, Transport Theory and Statistical Physics 1(2) (1971) 101–114.
- [6] R.G. Lord, Some Extensions to the Cercignani-Lampis Gas-surface Scattering Kernel, Physics of Fluids 3(4) (1991) 706–710.
- [7] Walter G. Vincenti, Charles H. Kruger, Jr., Introduction to Physical Gas Dynamics, John Wiley & Sons, 1965.
- [8] Ching Shen, Rarefied Gas Dynamics: Fundamentals, Simulations and Micro Flows, Springer, 2005.